



# ThaiSpoof: A Database for Spoof Detection in Thai Language


Kasorn Galajit

*NECTEC, National Science and Technology Development Agency,*  
Pathum Thani, Thailand  
kasorn.galajit@nectec.or.th 

Masashi Unoki

*Japan Advanced Institute of Science and Technology,*  
1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan  
unoki@jaist.ac.jp 


Pakinee Aimmanee

*Sirindhorn International Institute of Technology,*  
*Thammasat University,* Pathumthani, Thailand  
pakinee@siit.tu.ac.th 

Win Pa Pa

*University of Computer Studies, Yangon*  
Yangon, Myanmar  
winpapa@ucsy.edu.mm


Teeradaj Racharak

*Japan Advanced Institute of Science and Technology,*  
1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan  
racharak@jaist.ac.jp 


Hayati Yassin

*Faculty of Integrated Technologies,*  
*Universiti Brunei Darussalam,* Brunei Darussalam  
hayati.yassin@ubd.edu.bn


Thunpisit Kosolsriwiwat

*Sirindhorn International Institute of Technology,*  
*Thammasat University,* Pathum Thani, Thailand  
6322772854@g.siit.tu.ac.th 


Candy Olivia Mawalim

*Japan Advanced Institute of Science and Technology,*  
1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan  
candyolim@jaist.ac.jp 


Waree Kongprawechnon

*Sirindhorn International Institute of Technology,*  
*Thammasat University,* Pathumthani, Thailand  
waree@siit.tu.ac.th 


Anuwat Chaiwongyen

*Japan Advanced Institute of Science and Technology,*  
1-1 Asahidai, Nomi, Ishikawa 923-1292 Japan  
anuwat@jaist.ac.jp 

Surasak Boonkla

*NECTEC, National Science and Technology Development Agency,*  
Pathum Thani, Thailand  
surasak.boonkla@nectec.or.th 

Jessada Karnjana

*NECTEC, National Science and Technology Development Agency,*  
Pathum Thani, Thailand  
jessada.karnjana@nectec.or.th 

**Abstract**—Many applications and security systems have widely applied automatic speaker verification (ASV). However, these systems are vulnerable to various direct and indirect access attacks, which weakens their authentication capability. The research in spoofed speech detection contributes to enhancing these systems. Unfortunately, the study in spoofing detection is limited to only some languages due to the need for various datasets. This paper focuses on a Thai language dataset for spoof detection. The dataset consists of genuine speech signals and various types of spoofed speech signals. The spoofed speech dataset is generated using text-to-speech tools for the Thai language, synthesis tools, and tools for speech modification. To showcase the utilization of this dataset, we implement a simple spoof detection model based on a convolutional neural network (CNN) taking linear frequency cepstral coefficients (LFCC) as its input. We trained, validated, and tested the model on our dataset referred to as ThaiSpoof. The experimental result shows that the accuracy of model is 93%, and equal error rate (EER)

is 6.78%. The result shows that our ThaiSpoof dataset has the potential to develop for helping in spoof detection studies.

**Index Terms**—Thai database, spoof detection, automatic speaker verification, speech synthesis, speech modification.

## I. INTRODUCTION

Automatic speaker verification (ASV) systems have gained widespread application in various fields and security systems. However, these systems are susceptible to direct and indirect access attacks, compromising their authentication capabilities. Significant research has been dedicated to spoofed speech detection to bolster the security of these systems. Unfortunately, the scope of this research is limited to some languages due to the need for appropriate datasets. This paper is a part of the ASEAN IVO 2023 project, “Spoof Detection for Automatic Speaker Verification,” which aims to enhance

the security and reliability of speaker verification by effectively detecting spoofing attacks.

In literature, in the beginning, voice spoofing detection research involved speech and speaker. An effective database and a performance metric was discussed at the INTERSPEECH 2013 special session on Spoofing and ASV countermeasures [1]. This first meeting raised Spoofing and Countermeasures Challenge, ASVspoof, in 2015. The dataset ASVspoof 2015 consists of two spoofing attacks: synthetic speech and voice conversion [2]. Consequently, ASVspoof 2017 [3], ASVspoof 2019 [4], and ASVspoof 2021 [5] were organized. Each ASVspoof challenge published an available dataset for download. There are several common publicly available datasets which are used by voice presentation attack detection researchers, for example, ReMASC [6], Spoofing and anti-spoofing (SAS) corpus [7], RedDots [8], Vox Celeb [9], voicePA [10], and BioCPqDPA [11]. All databases are in the English language. The only ADD dataset from the Audio Deepfake Detection Challenge is in the Chinese language [12].

This paper addresses the need for a Thai language dataset specifically designed for spoof detection. In prosodic features, intonation, and accent are most important in the Thai language. Therefore, the database in Thai is necessary for studying spoof detection of the Thai voice. This paper aims to contribute to the ongoing efforts to enhance ASV system security by providing a robust and reliable Thai language dataset for spoofed speech detection.

## II. DATABASE DEVELOPMENT

This section provides detail about database development. The dataset consists of the genuine dataset and the spoof dataset. The genuine dataset comprises 143,262 utterances developed from Common Voice Corpus 13 [13]. The spoof dataset consists of 1,575,882 utterances and is generated using different techniques. Three techniques: text-to-speech, fundamental frequency (F0) modification, and pitch shifting, are performed to provide a spoof dataset. The number of signals in the text-to-speech dataset equals the number of genuine speech signals. The number of signals in the F0 modification dataset is four times that of genuine speech signals because we vary the value of F0 with four different values. The number of signals in the pitch-shifting dataset is six times that of genuine speech signals because we apply six different values for pitch-shifting. Table I shows information about our ThaiSpoof database. The detail of generating each type of signal in the database is provided in this section.

### A. Genuine Dataset

This study’s genuine dataset was sourced from the Common Voice Corpus database [13]. Common Voice is a free and open-source voice dataset that anyone can contribute to. Datasets contain a diverse range of speakers and languages. The Common Voice Corpus contains 28,118 hours of recorded audio, including a corresponding text file. The audio files include demographic metadata such as age, sex, and accent, which can help train the accuracy of speech recognition

TABLE I  
A SUMMARY OF THAISPOOF DATABASE FOR SPOOF DETECTION.

Label	Database type	Degree	No. utterance
Genuine	Genuine dataset	-	143,262
	Text-to-speech dataset	-	143,262
Spoofed	F0 modification dataset	10 ch/oct	143,262
		40 ch/oct	143,262
		160 ch/oct	143,262
		320 ch/oct	143,262
	Pitch shifting	+4%	143,262
		+10%	143,262
		+20%	143,262
		-4%	143,262
		-10%	143,262
		-20%	143,262

engines. The dataset currently consists of 18,652 validated hours in 112 languages. The Thai Common Voice Corpus 13.0 dataset utilized in our study contains 416 recorded hour and 167 validated hour from 7,784 speakers. Note that only the validated dataset was employed for our genuine dataset. The original utterances are in MP3 format. We change its format to become wav file.

### B. Spoof Dataset

1) *Text-to-Speech Dataset*: Text-to-Speech (TTS) technology has undergone significant advancements, enabling the synthesis of speech that closely resembles human-like expressions from written text. Within this domain, VAJA 9.0, an open-source software developed by AI for THAI. This software effectively addresses three essential components of the TTS pipeline for the Thai language: text processing, text-to-phoneme conversion, and speech synthesis. The text-to-speech process is shown in Fig. 1.

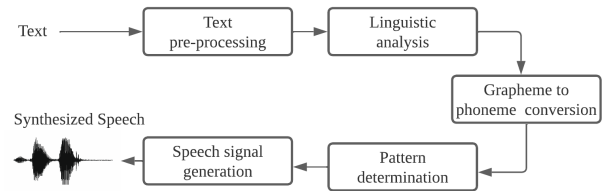


Fig. 1. The process of text-to-speech.

**Text Processing:** This technique prepares the input text for speech synthesis. It encompasses four sub-steps:

- Text pre-processing.* Managing punctuation, special characters, and whitespace ensure text uniformity. Normalization techniques standardize text representation.
- Linguistic Analysis.* After pre-processing, the text undergoes linguistic analysis to identify key units-words, syllables, and phonemes. VAJA 9.0 introduces pseudo syllables and smaller word segments that enhance pronunciation accuracy [14]. For example, "Hello, nice to meet you" becomes ['Hel', 'lo', 'ni', 'ce', 'to', 'meet', 'you'].
- Pseudo Syllable Generation.* Pseudo syllables are generated by considering the linguistic properties of individual characters, ensuring context-aware pronunciation aligned

with natural speech patterns. These encapsulate phonetic and phonological characteristics, leading to smoother and more natural-sounding synthesized speech.

- d) *Speech Signal Generation*. Pseudo syllables form the basis for speech signal generation. The TTS system converts linguistic representations into corresponding acoustic representations, incorporating pitch, duration, and amplitude. These representations are combined to create the final synthesized speech waveform.

**Text-to-Phoneme Conversion:** This technique accurately converts written text into corresponding phonemes, the basic unit of speech sounds. It consists of three sub-steps:

- a) *Sequence-to-Sequence (Seq2Seq) Learning*: The encoder processes input text and generates a context vector. The decoder then creates the phoneme sequence, while paired data helps the model learn the mapping between text and phonemes.
- b) *Conditional Random Fields (CRFs) for Phoneme Prediction*: CRFs are probabilistic graphical models that consider contextual dependencies between neighboring phonemes. Unlike Seq2Seq models, CRFs model the probability distribution over all possible phoneme sequences, given the input text [14]. CRFs perform joint inference over the entire sequence, accounting for interactions between neighboring phonemes. By capturing contextual dependencies, CRFs enhance the accuracy and naturalness of synthesized speech [15].
- c) *Integration of Sequence-to-Sequence and CRFs*: Seq2Seq and CRFs synergize in VAJA for precise phoneme prediction. Seq2Seq sets the foundation, and CRFs refine phoneme accuracy by considering the context.

**Speech Synthesis:** This technique involves transforming linguistic representation into acoustic features for speech generation. It consists of three sub-steps:

- a) *Transformation of Linguistic Representations*: Linguistic information, such as pseudo syllables and phonemes, is transformed into acoustic features. These features encode critical speech characteristics like pitch, duration, and amplitude.
- b) *Acoustic Representation to Speech Signal*: Acoustic features are combined to produce the speech signal waveform. The waveform captures intricate details for natural-sounding speech, including prosody, rhythm, and articulation.
- c) *Sound Synthesis*: The final synthesized sound waveform is created by seamlessly merging acoustic features. This waveform closely mimics human speech and effectively translates written text into expressive, intelligible speech.

In this study, we employed VAJA text-to-speech technology to generate high-quality speech. Leveraging the capabilities of VAJA, we seamlessly processed the text files sourced from the Common Voice Corpus 13.0, effectively transforming written text into spoofed-sounding speech with remarkable accuracy and fluency.

2) *Fundamental Frequency Modification Dataset*: A fundamental frequency (F0) of a person’s voice can give listeners clues about the speaker’s identity, gender, and age. Studies have shown that when the F0 of a voice is changed, for example, if a voice is made higher-pitched, people will remember it as being even higher-pitched than it was initially. Therefore, changing F0 can fault the automatic speaker verification system or the person [16]. We employ the WORLD vocoder, a free software tool, to analyze, manipulate, and synthesize speech. It can estimate the fundamental frequency (F0), aperiodicity, and spectral envelope of speech, and it can also generate speech that sounds like the input speech with only the estimated parameters [17]. The structure of WORLD vocoder is shown in Fig. 2.

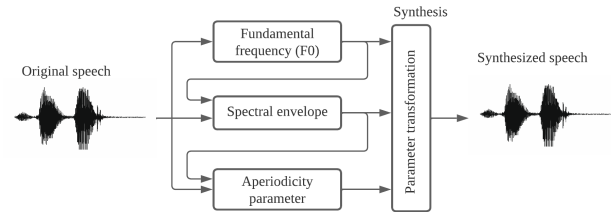


Fig. 2. Structure of WORLD vocoder.

There are several versions of WORLD. However, all versions apply the same main concept to decompose speech waveform into constituent parts, such as the fundamental frequency (F0), spectral envelope, and aperiodicity [18]–[20]. Users can transform this parameter to compose the synthesized speech. The version of WORLD used in this study applied a fundamental frequency estimator named *Harvest* [20]. The spectral envelope is estimated using the CheapTrick algorithm. Lastly, the D4C algorithm is applied for aperiodicity estimation [18]. This spoofed dataset focuses on F0 modification. Therefore, the concept of Harvest is described. Since continuous F0 modeling assigns a fixed F0 to unvoiced sections, Harvest tries to reduce the number of unvoiced frames and assign them more accurate F0 values. Harvest consists of two stages: estimating F0 candidates and generating a reliable F0 contour based on these candidates.

The purpose of the first stage of Harvest is to collect all F0 candidates even if they include estimation errors. The outline of the first stage is shown in Fig. 3. It consists of four sub-stages.

- a) *Estimation of the basic F0 candidates*. Since our spoofed speech was created from F0 modification, let us consider closely on the F0 candidate estimation. The speech waveform is filtered by many band-pass filters with different center frequencies. The filter  $h(t)$  is designed by multiplying the Nuttall window  $w(t)$  and the sine wave [21].

$$h(t) = w(t) \cos(\omega_c t), \quad (1)$$

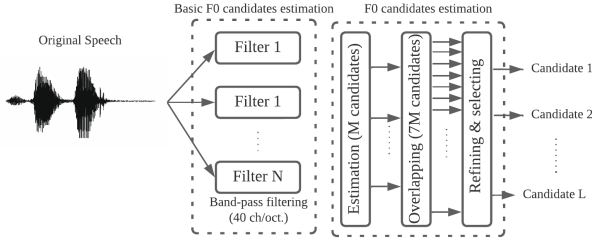


Fig. 3. Outline of the first stage of Harvest.

and

$$\begin{aligned}
 w(t) = & 0.355768 + 0.487396 \cos\left(\frac{\pi}{2T_c}t\right) \\
 & + 0.144232 \cos\left(\frac{\pi}{T_c}t\right) \\
 & + 0.012604 \cos\left(\frac{3\pi}{2T_c}t\right),
 \end{aligned} \quad (2)$$

where  $\omega_c$  represent the center frequency of the filter, and  $T_c$  is  $\frac{2\pi}{\omega_c}$ . The filter has range from  $-2T_c \leq t \leq 2T_c$ . The filter can extract the fundamental frequency of speech if the frequency is within a certain range near the filter's center frequency ( $\omega_c$ ). However, the fundamental frequency is unknown before it is estimated, so many filters with different center frequencies are needed. In Harvest, the center frequencies of the filters are assigned to a different number of a number of channels per octave (ch/oct) from the floor and ceiling frequencies to make the speech signal become spoofed speech. The number of ch/oct was set as follows, 10 ch/oct, 40 ch/oct, 160 ch/oct, and 320 ch/oct. The filter's output signal is shaped like a sine wave when only the fundamental component is extracted. The basic F0 candidate is calculated as the inverse of the average of the four intervals in the output signal. Any estimated candidate that is not included in the range of  $\omega_c \pm 10\%$  is removed.

- b) *Estimation of F0 candidates from basic F0 candidates.* Harvest obtains the F0 candidate when the filter outputs the same basic F0 candidates in a certain bandwidth. The bandwidth is set to  $\omega_c \pm 10\%$ .
- c) *Overlapping F0 candidates.* Sometimes, there are frames with no F0 candidates because of noise. One way to solve this problem is to overlap the F0 candidates from neighboring frames. Harvest overlaps the F0 candidates by  $\pm 3$  ms.
- d) *Refining and scoring all F0 candidates by instantaneous frequency.* Harvest refines and scores all F0 candidates by using the instantaneous frequency.

The purpose of the second stage is to create a single, accurate F0 contour from all of the possible F0 values. It consist of four sub-stages.

- a) *Removal of unwanted F0 candidates.* Harvest removes F0 candidates that change too quickly or that are outside of the expected frequency range for voiced speech.

- b) *Removal of short voiced sections.* Short voiced sections with a length below the threshold are removed and counted as the unvoiced section.
- c) *Expansion of each voiced section.* Harvest expands voiced sections by looking for F0 candidates in unvoiced sections. The expansion is limited to 100 ms, and short voiced sections are removed after expansion. If two expanded F0 contours overlap, the one with the higher reliability score is selected.
- d) *Interpolation and smoothing of the F0 contour.* F0s in this section are given by the linear interpolation between the F0s of the anteroposterior voiced section of their boundaries, then the connected F0 contour is smoothed in each voiced section by a zero-lag Butterworth filter. The smoothing result is the final F0 contour estimated by Harvest.

3) *Pitch-shift.* The pitch of a person's voice can be affected by several factors, including their vocal cords, their size and shape, and their breathing. These factors can also be affected by age, gender, and emotion. For example, young children typically have higher-pitched voices than adults, and women usually have higher-pitched voices than men. Therefore, if the pitch of speech signal is manipulated, The manipulated speech can fault the automatic speaker verification system or lead to missed speaker identification.

Since our spoofed speech was created by shifting the pitch of speech, we deployed a time-scale modification algorithm for speech using pointer interval control overlap and add (PICOLA) developed N. Morita *et al.* [22]. PICOLA is the method to adjust the length of utterance by expanding or condensing voiced vowels on the time sequence. This algorithm was derived from time domain harmonic scaling (TDHS). Thus, PICOLA has advantages by using just a period size buffer, making its code easy to implement.

Generally, time-scale modification (TSM) algorithms use a single speaking rate,  $r$ , for all time scales. This speaking rate,  $r$ , is a real number between 0 and 1 when speeding up the speech, and greater than 1 when slowing the speech. The speaking rate of the speech without modification is typically set to 1. Let us consider the slowing case as shown in shown in Fig. 4 (left).

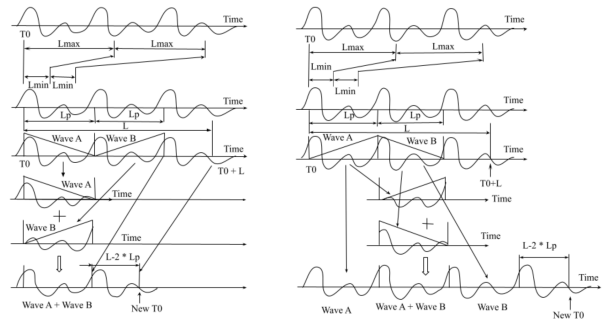


Fig. 4. Time scale modification: slowing down (left) and speeding up (right).

The algorithm separates a signal into the exact period size,

TABLE II  
SUMMARY OF IMPLEMENTED DATABASE.

Dataset	Degree	setA	setB	setC	setD(Dev)	Test	Total
Genuine	-	16,500	16,500	16,500	16,500	16,500	82,500
TTS	-	5,500	5,500	5,500	5,500	5,500	27,500
F0 changing	10 ch/oct	1,375	1,375	1,375	1,375	1,375	6,875
	40 ch/oct	1,375	1,375	1,375	1,375	1,375	6,875
	160 ch/oct	1,375	1,375	1,375	1,375	1,375	6,875
	320 ch/oct	1,375	1,375	1,375	1,375	1,375	6,875
Pitch shifting	+4%	917	917	917	917	917	4,585
	+10%	917	917	917	917	917	4,585
	+20%	917	917	917	917	917	4,585
	-4%	917	917	917	917	917	4,585
	-10%	916	916	916	916	916	4,580
	-20%	916	916	916	916	916	4,580
Total							165,000

$L_p$ , i.e., waves A and B in Fig.4(left). Then A + B is crossfaded with parameter  $r$ . This speech signal is compressed using rate  $r$ . Parameter  $L$  is a reduced length which can be determined using:

$$L = \left\lfloor \frac{L_p}{r - 1} \right\rfloor, \quad (3)$$

where  $\lfloor \cdot \rfloor$  is a round function.

Let us consider the speeding up case as shown in Fig.4(right). The algorithm separates a signal into the exact period size,  $L_p$ . Then A + B is combined with parameter  $r$ . This speech signal is expanded using rate  $r$ . Parameter  $L$  is a increased length which can be determined using:

$$L = \left\lfloor \frac{r}{1 - r} L_p \right\rfloor. \quad (4)$$

This speaking rate,  $r$ , is a parameter we adjust to various values to create the spoofed speech dataset. We create the pitch-shift dataset using six different values of the speaking rate: +4%, +10%, +20%, -4%, -10%, and -20%. Therefore, the speaking rate,  $r$ , was set to be 1.04, 1.10, 1.20, 0.96, 0.9, and 0.8, respectively.

### III. IMPLEMENTATION AND EVALUATION OF A SPOOFING DETECTION MODEL

To showcase the practical use of ThaiSpooF, we implemented a model that classifies a genuine and spoofed speech signal. Also, performance evaluation of the model is provides in this section.

#### A. Dataset

We select 165,000 utterances for this experiment, where 82,500 utterances is genuine and another half comprises various types of spoofing. The dataset for evaluation is shown in Table II.

#### B. Implementation

The framework for spoofing detection is shown in Fig. 5.

To demonstrate the use of our dataset, we implement a model based on a convolutional neural network (CNN) that classifies speech signals using their linear frequency cepstral coefficients (LFCCs). The details of feature extraction and model architecture are as follows.

LFCC is a feature typically used in many speech signal processing applications. The general process to compute LFCC

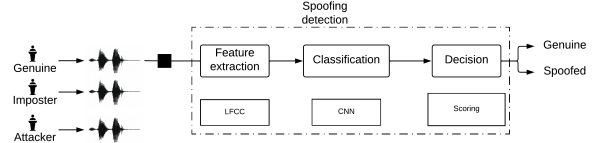


Fig. 5. Spoof detection framework.

is as follows. Firstly the input speech signal is segmented into overlapping frames. The frame length is 0.030 seconds, and the step between successive frames is 0.015 seconds. Each frame is weighted using the Hamming window, and then the discrete Fourier transform is performed on the weighted frame. A linearly-spaced filterbank with 70 filters is applied on those frames. The filterbank captures the energy distribution of the signal in different frequency bands. Then, the logarithmic function is taken to energies from the filterbank to compress the dynamic range. Finally, the discrete cosine transform is applied to the log spectrum energies, and the first 20 coefficients are used. In this work, LFCCs are used with their deltas and delta-deltas as the input of the classification model.

The classification model is a CNN consisting of five convolutional layers, one flatten layer, and three fully connected layers. The input LFCC dimension is 128-by-60, and there are two classes: genuine and spoofed. The numbers of filters and their sizes in the convolutional layers are 96 (with a size of 5-by-5), 256 (3-by-3), 384 (3-by-3), 384 (3-by-3), and 256 (3-by-3), respectively. All use the ReLU activation function. The outputs of the first, second, and fifth convolutional layers are pooled using max pooling with a pool size of 2-by-2 and a stride of 2-by-2. The output of the last convolutional layer is flattened and then forwarded to three fully connected layers, each with 4096 neurons, 50% dropout, and ReLU activation function. The output layer is a fully connected layer with two neurons, one for each class, and the softmax activation function is used to output a probability distribution between those two classes. The optimizer used for training the model is Adam, with a learning rate of 0.000177828, and the loss function is the sparse categorical cross-entropy.

#### C. Evaluation and Results

The performance of an a spoof detection system is typically measured using the equal error rate (EER), which is the point where the false accept rate (FAR) and false reject rate (FRR) are equal [23]. The EER is a widely used metric for anti-spoofing, and a lower EER value indicates better performance. Other classification metrics, such as accuracy, recall, and F1 score, are also used to evaluate a spoof detection systems. Table III shows the evaluation result of the model developed on ThaiSpooF.

### IV. DISCUSSION AND CONCLUSION

Based on the evaluation, this study delved into the development and evaluation of a CNN model tailored for

TABLE III

EVALUATION OF THE IMPLEMENTED MODEL ON OUR DATABASE.

4-fold cross-validation	Loss	Accuracy	Balanced Acc	Precision	Recall	F1	EER
Round1	Train	0.0927	0.9722	0.9722	0.9919	0.9522	0.9716
	Validation	-	0.9320	0.9320	0.9584	0.9033	0.9300
Round2	Train	0.0934	0.9623	0.9623	0.9939	0.9410	0.9668
	Validation	-	0.9773	0.9773	0.9843	0.9701	0.9771
Round3	Train	0.0954	0.9615	0.9615	0.9902	0.9524	0.9709
	Validation	-	0.9345	0.9345	0.9520	0.9152	0.9332
Round4	Train	0.0981	0.9610	0.9610	0.9745	0.9709	0.9727
	Validation	-	0.9302	0.9302	0.9350	0.9247	0.9298
	Test	-	0.9322	0.9322	0.9509	0.9114	0.9307

the detection of spoofed speech, Notably, the model achieved an impressive accuracy of 93.22% during the final epoch of training. This remarkable accuracy underscores the primary objective of enhancing the security and dependability of ASV systems.

During the evaluation phase, the model's robustness was assessed on the development and test dataset. Impressively, the test dataset yielded accuracy, balanced accuracy, precision, recall, and F1-score of 93.22%, 93.22%, 95.09%, 91.14%, 93.07% respectively, with the ERR of 6.78%. These outcomes emphasize the model's reliability in distinguishing between genuine and spoofed speech.

The achieved results not only demonstrate the proposed LFCCs and CNN classification but also contribute to the broader goal of advancing spoofed speech detection. By effectively addressing the challenges posed by a diverse dataset containing various forms of spoofed speech, this study contributes to improve the efficiency of ASV system.

In conclusion, the result shows that this ThaiSpoof dataset has the potential to develop for helping in spoof detection studies, and this study paves the way for future investigations aimed at bolstering the security of ASV systems in the context of the Thai language. As the field of spoof detection continues to evolve, this research not only fills a critical gap in spoof detection for the Thai language but also provides a foundation for further advancements. By presenting a comprehensive dataset and a model framework that effectively addresses the challenges of Thai language prosody and accent, this study contributes to the ongoing efforts to fortify the security landscape of ASV systems.

#### ACKNOWLEDGMENT

The ASEAN IVO ([http://www.nict.go.jp/en/asean\\_ivo/index.html](http://www.nict.go.jp/en/asean_ivo/index.html)) project, "Spoof Detection for Automatic Speaker Verification", was involved in the production of the contents of this presentation and financially supported by NICT (<http://www.nict.go.jp/en/index.html>).

#### REFERENCES

- [1] N. W. Evans, T. Kinnunen, and J. Yamagishi, "Spoofing and countermeasures for automatic speaker verification." in *Interspeech*, 2013, pp. 925–929.
- [2] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Haniçli, M. Sahidullah, and A. Sizov, "Asvspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge," in *Sixteenth annual conference of the international speech communication association*, 2015.
- [3] T. Kinnunen, M. Sahidullah, H. Delgado, M. Todisco, N. Evans, J. Yamagishi, and K. A. Lee, "The asvspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," 2017.

- [4] A. Nautsch, X. Wang, N. Evans, T. H. Kinnunen, V. Vestman, M. Todisco, H. Delgado, M. Sahidullah, J. Yamagishi, and K. A. Lee, "Asvspoof 2019: spoofing countermeasures for the detection of synthesized, converted and replayed speech," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, pp. 252–265, 2021.
- [5] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans *et al.*, "Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection," *arXiv preprint arXiv:2109.00537*, 2021.
- [6] Y. Gong, J. Yang, J. Huber, M. MacKnight, and C. Poellabauer, "Remasc: realistic replay attack corpus for voice controlled systems," *arXiv preprint arXiv:1904.03365*, 2019.
- [7] Z. Wu, A. Khodabakhsh, C. Demiroglu, J. Yamagishi, D. Saito, T. Toda, and S. King, "Sas: A speaker verification spoofing database containing diverse attacks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4440–4444.
- [8] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmner, D. Van Leeuwen, H. Aronowitz, M. Kockmann, C. Vaguero, B. Ma *et al.*, "The reddit data collection for speaker recognition," in *Interspeech 2015*, 2015.
- [9] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.
- [10] P. Korshunov and S. Marcel, "A cross-database study of voice presentation attack detection," *Handbook of Biometric Anti-Spoofing: Presentation Attack Detection*, pp. 363–389, 2019.
- [11] P. Korshunov, A. R. Gonçalves, R. P. Violato, F. O. Simões, and S. Marcel, "On the use of convolutional neural networks for speech presentation attack detection," in *2018 IEEE 4th international conference on identity, security, and behavior analysis (ISBA)*. IEEE, 2018, pp. 1–8.
- [12] J. Yi, J. Tao, R. Fu, X. Yan, C. Wang, T. Wang, C. Y. Zhang, X. Zhang, Y. Zhao, Y. Ren *et al.*, "Add 2023: the second audio deepfake detection challenge," *arXiv preprint arXiv:2305.13774*, 2023.
- [13] Mozilla Common Voice, "Common voice," 2023, <https://commonvoice.mozilla.org/en/datasets>, Last accessed on 2023-08-03.
- [14] C. Wutiw WATCHAI, C. Hansakunbuntheung, A. Rugchatjaroen, S. Saychum, S. Kasuriya, and P. Chotrakool, "Thai text-to-speech synthesis: a review," *Journal of Intelligent Informatics and Smart Technology*, vol. 2, no. 2, pp. 1–8, 2017.
- [15] A. Rugchatjaroen, S. Saychum, S. Kongyoung, P. Chotrakool, S. Kasuriya, and C. Wutiw WATCHAI, "Efficient two-stage processing for joint sequence model-based thai grapheme-to-phoneme conversion," *Speech Communication*, vol. 106, pp. 105–111, 2019.
- [16] G. E. Gous, *Effects of manipulating fundamental frequency and speech rate on synthetic voice recognition performance and perceived speaker identity, sex, and age*. Nottingham Trent University (United Kingdom), 2017.
- [17] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.
- [18] M. Morise, "D4c, a band-aperiodicity estimator for high-quality speech synthesis," *Speech Communication*, vol. 84, pp. 57–65, 2016.
- [19] M. Morise and Y. Watanabe, "Sound quality comparison among high-quality vocoders by using re-synthesized speech," *Acoustical Science and Technology*, vol. 39, no. 3, pp. 263–265, 2018.
- [20] M. Morise, "A high-performance fundamental frequency estimator from speech signals," in *Proc. INTERSPEECH*, pp. 2321–2325.
- [21] A. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 1, pp. 84–91, 1981.
- [22] N. Morita, "Time-scale modification algorithm for speech by use of pointer interval control overlap and add (picola) and its evaluation," *Proc. ASJ*, pp. 149–150, 1986.
- [23] Z. Wu, J. Yamagishi, T. Kinnunen, C. Haniçli, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "ASVspoof: the automatic speaker verification spoofing and countermeasures challenge," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 4, pp. 588–604, 2017.